International Journal of Bifurcation and Chaos, Vol. 7, No. 3 (1997) 751-757

ADAPTIVE EVOLUTIONARY OPTIMIZATION OF TEAM WORK

VOLKHARD BUCHHOLTZ and THORSTEN PÖSCHEL
Institut für Physik, Humboldt–Universität zu Berlin,
Invalidenstr. 110, D–10115 Berlin, Germany

Received March 7, 1996
Revised August 12, 1996

We discuss a new optimization strategy, which considerably improves the effectivity of evolutionary algorithms applied to a certain class of optimization problems. The basic principle is to solve first a simpler related problem, which is constructed by introducing additional degrees of freedom to the landscape. Starting from the solution in this simplified landscape we remove stepwise the added degrees of freedom. Our optimization strategy is demonstrated for a sample problem.

1. Introduction

Evolutionary algorithms have been shown by many authors (e.g. [11]) to be suited for solving complex optimization problems, and there are a highly developed theory of evolution processes [6] and lots of recipes for successfully applying evolutionary algorithms [21]. Particularly in combinatoric optimization where many other standard techniques fail, considerable success has been achieved by using evolutionary algorithms. Obviously, each particular optimization problem could be solved more effectively (i.e. less computer time consuming) by a deterministic algorithm, provided the algorithm is known. The main advantage of evolutionary algorithms is their simplicity and universality. In most cases evolutionary algorithms can be parallelized in a trivial way by distributing the individuals among a set of parallel processors. This may be considered as another advantage.

One of the most well-known examples, which has drawn the attention of scientists from various disciplines, is the Travelling Salesman Problem (TSP). In its standard formulation, the TSP has been proven to be NP complete [8], and many authors have tried to tackle this problem using different types of evolutionary strategies as well as other methods (e.g. [19,2,3,10,23,24]).

To apply an evolutionary algorithm one has to provide two essential ingredients: First, one needs a fitness function which evaluates a solution. This fitness function must be calculable very effectively (in terms of computer time) since it has to be evaluated extremely often during the calculation. Hence, the fitness function is required to be simple. Second, one needs a mutation operator which takes into account the topology of the fitness as a function of the parameters to be optimized. If the mutation operator does not care about the topology it happens that even a small mutation may lead to an extreme change in the fitness, and hence, the evolution algorithm turns into stochastic search. Examples can be found in the literature where either the first or the second precondition is violated and where the evolutionary algorithm does not work effectively.

In the present paper we investigate an evolutionary game where each individual is a set of points, i.e. a team, which solves a well defined problem by cooperative behavior, i.e., by "team work". In the beginning neither the number of points per team N_{min} which is necessary to solve the problem nor the detailed solution is known. Both have to be found during the evolutionary game. Formally, one can consider the solution to be a subspace of the configuration space of dimension 2 N_{min}.

It will be shown that even if we knew the number

of necessary points it would be favourable to solve first a simpler problem, in which the teams consist of more points than necessary, and then stepwise to increase the complexity of the problem by reducing the number of points. Solving such a hierarchy of problems and using the solution of a simple problem as the initial condition for the next difficult problem can be much more effective.

2. Description of the Problem

Assume we have a complicated shaped room with polygonal ground-plan. The M even walls of the room have to be illuminated completely using a set of N light bulbs. The questions which will be investigated here are: *How many light bulbs are needed at minimum to illuminate the walls of the room, and where to place them?*

Although we cannot provide a proof for NP-completeness of the problem, probably most people will agree that this problem is complex in the general case. There is a short proof for the upper limit of bulbs needed to illuminate a room bounded by M even walls [7]. Provided that the room does not have columns one does not need more than $N = \lfloor M/3 \rfloor$ bulbs, where $\lfloor a \rfloor$ denotes the integer of a. Figure 1 displays a room where one needs indeed $N = \lfloor M/3 \rfloor$ bulbs. In many cases, however, significantly less than $N = \lfloor M/3 \rfloor$ are necessary for complete illumination. We want just to note that for the case that the room has S inner columns (of polygonal cross section) one claims ([18]) that $N = \lfloor (M + S)/3 \rfloor$ lamps are needed, where M includes the number of walls of the room and of the columns. So far, however, there is no proof.

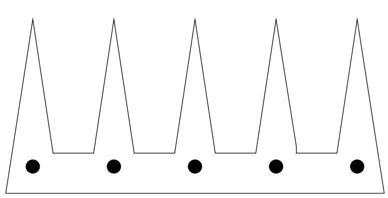

Fig. 1. In the drawn case of a room with 15 walls we find the worst case where $N = \lfloor 15/3 \rfloor = 5$ light bulbs will be needed to illuminate the walls completely. The dots show the positions of the bulbs.

A related problem is the so called *gallery watchman problem*[5]. The question here is how many (static) watchmen are needed to watch the walls of a museum (e.g. [22]), or in the dynamic formulation what is the shortest path of a watchman to pass through all points of interest. There are many formulations of the watchman problem and much theoretic work has been done in this field (e.g. [18,14,4,16,15]). A technical application of a one dimensional version of the dynamic watchman problem solved by an evolutionary algorithm was recently investigated by Heckman [9]. A machine consisting of a number of pickers assembled in a linear array across a conveyor belt was optimized to pick up pieces which move on the conveyor. The algorithm had to decide which of the pickers picks up the next coming piece.

In the case of our problem we assume that the room has no columns, therefore the number N_{min} of actually needed lamps is always less than the third of the number of walls $N_{min} \leq \lfloor M/3 \rfloor$. The fitness F of an individual, i.e., of a set of light bulbs, is given by the illuminated area*of all walls

$$F_i = \frac{\sum\limits_{i=1}^{M} k_i}{\sum\limits_{i=1}^{M} l_i}, \tag{1}$$

where k_i is the illuminated area of the ith wall and l_i is its total area. Hence we find for the fitness $F \in [0, 1]$. During the optimization we try to maximize F by changing the positions of the light bulbs. The solution is found when we have determined the positions of the minimum of a number of bulbs to illuminate the room, i.e. $F = 1$. The function F is embedded in the $2N$ dimensional space of the coordinates of the N bulbs. It may have a complex topology, in particular numerous discontinuities, local extrema and flat plateaus. Hence a simple gradient strategy would fail to find the optimum and one has to chose a more sophisticated optimization strategy. In light of the generally used classification scheme introduced by Schwefel [20] our algorithm is of (μ, λ)–type.

Each individual α, $(\alpha = 1 \ldots \mu)$ in our evolutionary game is a set of $N_{max} \geq N \geq N_{min}$ light bulbs, where $N_{max} = \lfloor M/3 \rfloor$ and N_{min} is the (unknown) minimal number of bulbs which are needed

*Since the problem is two dimensional we may call the length of a certain part of the border of the room area.

to illuminate the room. The algorithm starts with μ of such individuals, the positions $\vec{S}_i^\alpha$, $i = 1 \dots N$ of the lamps are random. The optimization scheme reads as follows:

1. The individuals are mutated by varying the position of each lamp
$$\vec{S}_i^\alpha \rightarrow \vec{S}_i^\alpha + m_w \cdot \vec{A}_i^\alpha, \tag{2}$$
with $i \in [1, N]$ and $\alpha \in [1, \mu]$. The components of the two dimensional vector $\vec{A}_i^\alpha$ are chosen equally distributed from the interval $[0, 1]$. The mutation step length m_w is constant. To avoid that the individuals persist in deep local maxima, with a small frequency P the new position of a lamp is completely random.

2. All individuals are rated by means of the fitness function (1).

3. If no set $\alpha \in [1 \dots \mu]$ solves the problem, i.e. $F_\alpha \neq 1$, the λ sets of lamps (individuals) with the highest fitness values are copied and brought over to the next generation. The remaining $\mu - \lambda$ individuals die out. Steps 1 to 3 are to be repeated until at least one of the individuals has the fitness $F_\alpha = 1$.

4. If at least one individual has fitness $F = 1$, the solution is found and we start to solve the next difficult problem, i.e., we try to solve the same optimization problem, now with each individual consisting of $N - 1$ lamps only. The solution of the previous problem with N lamps is used to initialize the new positions of $N - 1$ lamps per individual: The winning individual with fitness $F = 1$ is copied μ times and for each individual one randomly chosen lamp is removed. Then we start the new optimization run with $N-1$ bulbs beginning with item 1.

This procedure is continued until no solution can be found anymore. The last solution, found by the algorithm will be assumed to be the solution of the optimization problem stated at the beginning of this paragraph.

3. Results and discussion

The proposed algorithm was applied to illuminate a room ($M = 82$) with the ground-plan shown in Fig. 2. The solution given by $N_{min} = 10$ crosses was found during an evolutionary game of $\mu = 60$ individuals. The optimization parameters were $\lambda = 5$, $P = 10^{-3}$ and $m_w = \min(x_{max}, y_{max}) \cdot 10^{-2}$ where x_{max} and y_{max} denote the maximum extent of the room in the x- and y-directions respectively.

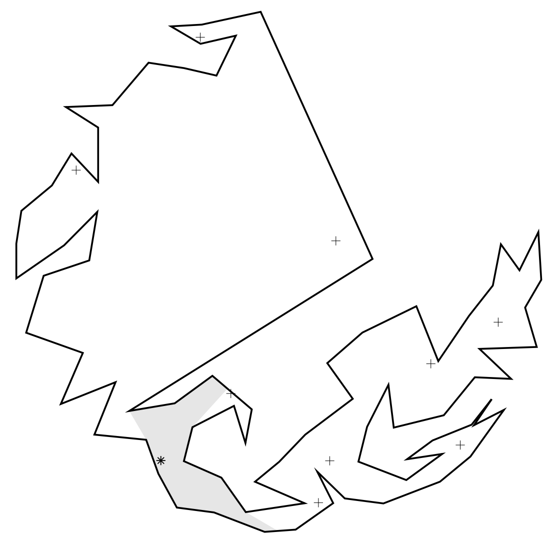

Fig. 2. Solution with $N_{min} = 10$ lamps for an arbitrary created room with $M = 82$ walls. The position of the lamps are drawn with (+)- and (*)-symbols. The room is completely illuminated, one can easily check that each place of the wall can be connected by a straight line with at least one lamp without intersecting the wall. The shadowed area displays the section which is illuminated by the lamp which position is given by the (*)-symbol.

The parameters were chosen to give a satisfying efficiency of the optimization for a much simpler room. We did not further optimize these parameters, because the aim of the paper is to present a new optimization scheme but not to improve the efficiency of the well-known evolutionary algorithm. For the optimization of the parameters see e.g. ref. [21].

In Fig. 3 the fitness F_I of the fittest individual I is plotted versus the number of evolutionary steps. We start up with random sets of $N = N_{min} = 10$ lights. The fitness is not

monotonously increasing in time but there are long periods of stagnation interrupted by rapid jumps in the fitness. (Note that the abscissa is drawn in log scale.) This behavior, which seems to be typical for evolutionary processes was observed by several authors for various problems before, e.g. [17], and substantiated theoretically (see [6]).

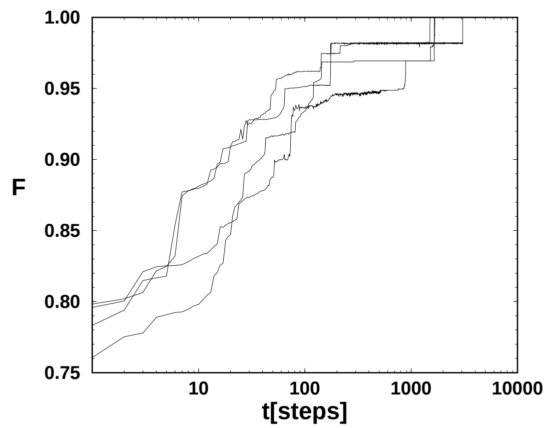

Fig. 3. Time evolution of the fitness F of the best individual for four runs starting with different initial conditions. There are long periods of nearly constant fitness interrupted by short periods of rapid fitness increasing.

Usually *a priori* we do not know the number N_{min} of lamps needed to illuminate the room and we start the optimization procedure with $N_0 > N_{min}$ lamps per individual. But even if we would know the number it would be favourable to start with a larger amount of lamps than needed. In the following we will show and explain, that the algorithm for our optimization problem is up to about 10 times faster for that case.

To ensure approximately the same amount of computer time for each evolutionary step unaffected by the number of lamps each set consists of, the number of individuals μ was chosen

$$\mu = \mu_{max} \cdot \frac{N_{min}}{N_0}, \tag{3}$$

where $\mu_{max} = 60$ is the number of individuals for an optimization starting with $N_0 = N_{min}$ light bulbs per individual. Hence, we can identify the number of evolutionary steps with time when we assume that the computer time needed for each step is mainly determined by the time required to calculate the fitness, i.e., it is proportional to the total amount of light bulbs.

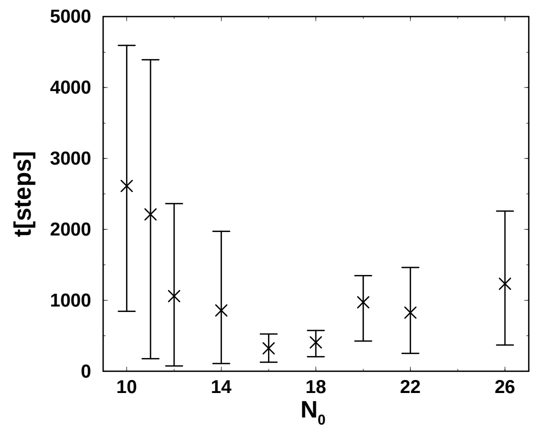

Fig. 4. Number of optimization steps t to find the final solution versus the number of bulbs N_0 in the starting iteration. The computer time which is proportional to the number of iteration steps decreases with increasing N_0 and reaches a minimum at $N_0 = 16$. For larger N_0 the value increases slowly.

Fig. 4 shows that the algorithm is in average about 10 times faster if one starts with sets consisting of $N_0 = 16$ lamps instead of the required $N_{min} = 10$. Each data point in this figure is the mean value of the needed optimization steps of five runs with different starting configurations. The data show that the time needed to compute the solution decreases with increasing starting number of lamps for $N_0 < 16$. When we start with larger sets $N_0 > 16$ the computer time increases slowly. There are two mechanisms responsible for this behavior:

1. Suppose we would do stochastic search, and suppose there would be a unique solution for the optimization problem. If we assume that τ is the time a single searcher needs to find one particular of the N_{min} places then N random independent searchers need the time

$$\tau^* = \frac{\tau}{N N_{min}} \tag{4}$$

to find *any* of the places. The aspect of team work is included by assuming that a searcher who has found "its place" will not

leave this place until the solution of the problem is found: It will survive the following evolutionary steps. Then we find for the time to find all places, i.e. to solve the optimization problem

$$T(N) = \sum_{i=0}^{N_{min}-1} \frac{\tau}{(N-i)\,(N_{min}-i)} \ . \qquad (5)$$

By replacing the sum by an integral we get the analytic solution

$$T(N) \approx \tau \int_{\frac{1}{2}}^{N_{min}-\frac{1}{2}} \frac{1}{(N-x)\,(N_{min}-x)} dx = \frac{\tau \ln\left(\frac{2N-1}{(2N_{min}-1)(2N-2N_{min}+1)}\right)}{N_{min}-N} \ . \qquad (6)$$

Obviously $T(N)-T(N+1)$ is a positive number, i.e. the solution will be found quicker when starting with $N+1$ lamps instead of N.

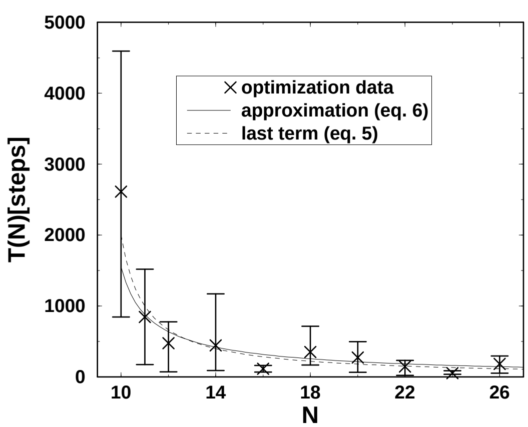

Fig. 5. Number of optimization steps T_N versus the number of lamps N. The averaged data over five runs of the evolution algorithm are drawn by crosses and error bars. The full line and the dashed line show the estimates due to eq. (5) and the last term of the sum in eq. (5) respectively. The time τ has been assumed to be a fit parameter.

Apart from this simple estimate we realize that typically the "attractor regions" for the positions of the bulbs do not have the same size but their sizes differ significantly. Usually the positions with the smallest "attractors" are found at the end of the optimization procedure, and the time is mainly determined by the last term of the sum in eq. (5). To provide better estimate one needs knowledge about the fitness landscape of the problem. For very simple problems it has been shown that one can conclude the properties of convergence based on knowledge od statistical properties of the fitness landscape [1].

Fig. 5 shows the results from the optimization as well as the discussed estimates.

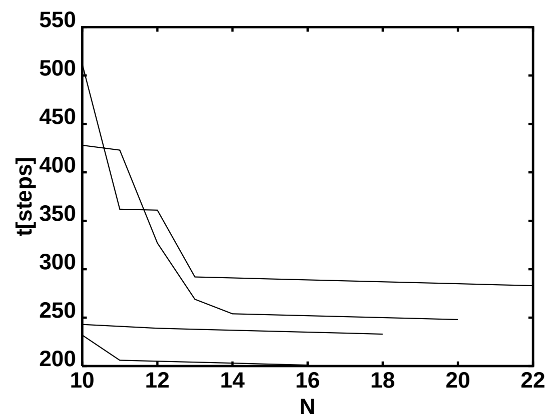

Fig. 6. Sample optimization runs. After finding a solution for $N_0 > N_{min}$ lamps the algorithm finds quickly the solution for smaller N. In accordance with fig. 4 the algorithm becomes slightly less effective when starting with larger N_0 (Explanation see text).

2. Fig. 6 shows the progress of the optimization procedure for $N_0 = \{16, 18, 20, 22\}$ lights in the starting configuration. One should read the figure from right to left: the ordinate shows the number of evolution cycles which have been necessary to solve the problem for N lamps. In the case of the run started with $N_0 = 18$ light bulbs the solution for $N = N_{min}$ was found almost exclusively by removing lights.

   Fig. 4 shows that the optimization time rises slightly for large N_0. Comparing the solutions of one optimization run for different N we note that for $N_0 \lesssim 18$ the initial solution (for $N = N_0$) is frequently very close to the

final solution (for $N = N_{min}$). Hence, the solution for $N-1$ lamps can be found from the solution for N lamps just by deleting one of the lamps and performing few evolution steps (see Fig. 6).

For $N_0 \gtrsim 18$ the initial solution could differ very much from the final solution, and also the solutions of the successive problems for N_0, $N_0 - 1 \ldots, N_{min}$ differ from each other. The solutions are not really adapted to the geometry of the room, but they are more the results of a random search. Therefore the system does not take too much advantage from knowing the solution for N lamps solving the problem for $N-1$ lamps. Hence the calculation time in Fig. 4 rises slightly for larger values of N. Nevertheless the algorithm is still much faster compared with the time needed for the solution starting with $N = N_{min}$ bulbs.

The situation is quite similar to a neural network applied to a pattern recognition problem: if it has too many neurons the network just stores the patterns instead of finding characteristic features (e.g. [12]). If this network is applied to an unknown pattern it fails since none of the stored patterns has enough overlap with the unknown pattern. A network with less neurons might be able to solve the problem since it checks whether the unknown pattern reveals characteristic features.

4. Conclusion

We have shown that for the investigated problem it is favourable to start the optimization with more points than necessary $N_0 > N_{min}$. First we find a solution, i.e. the positions of the points for a higher number of points, and then we stepwise decrease N while using the solution for N as initial condition for finding the solution for $N-1$ points. It turns out that the procedure to solve the chain of problems for $N_0, N_0-1, N_0-2, \ldots, N_{min}$ is up to about ten times faster then to solve the problem for N_{min} directly. Formally we first solve the optimization problem in a high dimensional space of dimension $2N_0$. Once the solution is found we stepwise reduce the dimensionality and end up with dimension $2N_{min}$.

The theoretical basis of the optimization speedup in higher dimension is provided by Morse theory [13]. Qualitatively Morse Lemma says that under rather mild assumptions the fraction of saddle points of a function of its critical points rises with dimension. Hence the relation of extrema and saddle points decreases when increasing the dimension. The idea of the lemma becomes clear in one and two dimensions: to reach the optimum of a one dimensional function one has to “walk through” all local extrema in between the starting point and the global extremum. In two dimensions in many cases one can “walk around” local extrema and hence one avoids time consuming escape procedures. Similar in higher dimension: the higher the dimension the more bypasses do exist to reach a certain point without getting stuck in local extrema.

We assume that this behavior is typical for a certain class of optimization problems. The main characteristic of this class is that the problem can be transformed into another problem containing a parameter N where the solution of the original problem with the parameter being N^* is contained in the set of solutions of the easier to solve problems with $N \neq N^*$. Obviously the entire class of coverage problems belongs to this class. Further investigations will be necessary to substantiate this hypothesis. For the problem of the travelling salesman, for instance, the easier task could be to find a minimal length where some cities (N) are allowed to be visited twice, followed by a successively reduction of the number of those cities.

The authors thank T. Asselmeyer, W. Ebeling, H. Herzel, G. Rudolph and L. Schimansky-Geier for stimulating discussions. V.B. thanks project EVOALG (BMBF) for financial support.

5. References

1. Asselmeyer, T. & Ebeling, W. [1996] “Mixing of Thermodynamical and Biological Strategies in Optimization”, Proceedings “International Twin-Conference on Complexity and Self-Organization” Berlin 1995 (in print).
2. Boseniuk, T. & Ebeling, W. & Engel, A. [1987] “Boltzmann and Darwin Strategies in Complex Optimization” *Phys. Lett.* **A 125**, 307-310.
3. Boseniuk, T. & Ebeling, W. [1988] “Evolution Strategies in Complex Optimization: The Travel-

ling Salesman Problem" *Syst. Anal. Model. Simul.* **5**, 413-422.

4. Chin, W-p. & Ntafos, S. [1988] "Optimum Watchman Routes" *Inf. Proc. Lett.* **28**, 39-44.
5. Chvàtal, V. [1975] "A Combinatorial Theorem in Plane Geometry" *J. Combin. Theory B* **13**, 395-398.
6. Ebeling, W. & Engel, A. & Feistel, R. [1990] *Physik der Evolutionsprozesse* (Akademie-Verlag, Berlin).
7. Fisk, S. [1978] "A short proof of Chvatal's watchman theorem" *J. Combin. Theory B* **24**, 374.
8. Garey, M. R. & Johnson, D. S. [1979] *Computers and Intractability: a guide to the theory of NP-completeness* (W. H. Freeman, San Francisco).
9. Heckman, T. [1995] "A Unique Application of Genetic Algorithms to Machine Control" (preprint).
10. Hopfield, J. J. & Tank, D. W. [1985] "'Neural' computation of decisions in optimization problems" *Biol. Cyber.* **52**, 141–152.
11. Goldberg, D. E. [1989] *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading).
12. Hertz, J. & Krogh, A. & Palmer, R. G. [1991] *Introduction to the Theory of Neural Computation.* (Addison-Wesley, Reading).
13. Milnor, J. [1963] *Morse theory* (Princeton Univ. Press, Princeton).
14. Ntafos, S. [1986] "On gallery watchmen in grids" *Inf. Proc. Lett.* **23**, 99-102.
15. Ntafos, S. & Chin, W-p. [1991] "Shortest Watchman Routes in Simple Polygons" *Discrete and Comp. Geom.* **6**, 9.
16. Ntafos, S. [1992] "Watchman Routes under Limited Visibility" *CGTA: Computational Geometry: Theory and Applications* **1**, 149.
17. Rechenberg, I. [1973] *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* (Frommann-Holzboog, Stuttgart-Bad Cannstatt).
18. O'Rourke, J. [1987] *Art Gallery Theorems* (Oxford Univ. Press, Oxford).
19. Reinelt, G. [1994] "The Travelling Salesman. Computational Solutions for TSP Applications", *Lecture notes in computer science*, **840** (Springer, Berlin).
20. Schwefel, H.-P. [1981] *Numerical Optimization of Computer Models* (John Wiley & Sons, Chichester, UK).
21. Schwefel, H. P. [1995] *Evolution and optimum seeking* (Wiley-Interscience, New-York).
22. Stewart, I. [1994] "How many Guards in the Gallery" *Scient. Am.* **270**, 88-92.
23. Whitley, D. & Kauth, J. [1988] "Genitor: A Different Genetic Algorithm" *Proceedings of the 1988 Rocky Mountain Conference on Artificial Intelligence*, (Denver), pp. 88-101.
24. Whitley, D. & Starkweather, T. & Shaner, D. [1991] "The Traveling Salesman and Sequence Scheduling : Quality Solutions Using Genetic Edge Recombination" in: L. Davis *Handbook of genetic algorithms* (Van Nostrand Reinhold, New York), pp. 350-371.